

## Общая характеристика работы

**Актуальность работы.** Сейчас существует много программ трехмерной визуализации, которые используют для реалистичного синтеза различные алгоритмы трассировки лучей. Эти программы применяются в кино- и телеиндустрии, дизайне, архитектуре и т.п. Сегодня наибольшей популярностью пользуются платные пакеты, такие как Maxwell Render, Lumion, Solidworks Visualize, Marmoset Toolbag, Corona Render, Iray, V-Ray, Indigo Render. Для независимых разработчиков и некоторых молодых компаний значительный расход средств на их приобретение может быть критическим, поэтому они склонны искать бесплатные аналоги. К счастью, такие существуют, но тогда встает вопрос о качестве их работы, ведь помимо цены, важнейшими характеристиками являются время рендеринга и фотореалистичность получающегося изображения. Важный фактор, влияющий на время – реализация распараллеливания алгоритмов рендеринга. С позиций требуемой вычислительной мощности процессоров также актуально рассмотреть и реализацию многопоточности у этих алгоритмов. Таким образом, *актуальность* заключается в том, что в работе сравниваются эти характеристики у некоторых популярных бесплатных рендеров и исследуется их оптимальная применимость к визуализации различных световых явлений.

**Цель работы** – сравнение реализованных алгоритмов трассировки лучей, использующихся в различных бесплатных программах для трехмерной визуализации, по качеству получаемого изображения относительно времени, затраченного на рендеринг.

**Научная новизна исследования** определяется тем, что на основе прямого практического применения рендеров исследуются их возможности и возможности алгоритмов трассировки, реализованных в них. Научно-практическая значимость результатов, заключается в возможности выбора потенциальным пользователем необходимого под его нужды конкретного бесплатного программного обеспечения. Также выводы, сделанные из результатов исследования, помогают понять и классифицировать алгоритмы по применимости к различным видам сцен и моделей.

На защиту выносятся следующие положения:

1. Построены и настроены идентичные экспериментальные модели разных рендеров для сравнения с эталонным изображением.
2. Получены графики, отображающие различие между эталонным изображением и экспериментальным в зависимости от времени и графики, отображающие процентное соотношение скорости при 6 потоках и при 12.

Достоверность результатов и выводов выпускной работы обеспечивается корректностью проведенных расчетов; подтверждается соответствием результатов, полученных разными методами, в тех случаях, когда это возможно, и согласием полученных результатов с выводами других исследователей в некоторых частных случаях.

#### **Апробация работы.**

Материалы ВКР докладывались на 69-й (2020) Научно-технической конференции РТУ МИРЭА.

#### **Публикации.**

По материалам исследования готовится к публикации статья «Производительность алгоритмов трассировки лучей на реалистичных сценах» в Российском технологическом журнале.

Личный вклад автора. Все изложенные в диссертации результаты получены автором лично. Автор принимал активное участие в обсуждении, интерпретации полученных результатов и написании статей. Вклад соискателя в опубликованные работы, вошедшие в диссертацию, является решающим. Структура и объем диссертации. Выпускная работа состоит из введения, трех глав, заключения, библиографии из 38 наименований и приложения. Работа изложена на ? страницах, содержит ? рисунков. Библиография включает 38 наименований на 4 страницах.

#### **Содержание работы**

**Во введении** отражена актуальность работы, указана цель проводимых автором исследований, кратко изложено содержание работы по главам.

**В первой главе** описываются теоретическая составляющая реалистичной визуализации изображений: физические законы, лежащие в основании теории физически-корректного рендеринга, сама теория. Также описывается аппарат, используемый для сравнения.

**Во второй главе** содержится подробное описание проводимых экспериментов: материалы, алгоритмы и их параметры. А также параметры получаемых в результате испытаний изображений.

**В третьей главе** приводятся графики, построенные на основании полученных данных. Сделаны выводы о возможности распараллеливания и скорости работы алгоритмов.

**В приложения** вынесен вспомогательный материал о реализации алгоритмов, а также эталонные изображения, используемые для сравнения.

В заключении сформулированы основные результаты и выводы работы:

1. Все рендеры улучшают скорость работы на ~66% при увеличении кол-ва потоков вдвое

2. appleseed показывает наибольшую реалистичность в имитации материалов, но является наиболее медленным из всех.

### **Слайд “Цели работы”**

Добрый день. Целью моей работы является сравнение нескольких рендеров, те программ для отрисовки изображений, и алгоритмов трассировки, реализованных в них. Проведя эксперименты мною были сделаны некоторые выводы о скорости работы этих алгоритмов и о реалистичности изображений, полученных в результате применения этих алгоритмов.

*Чтобы* провести корректное сравнение мне во-первых, необходимо было смоделировать различные сцены. В них должны быть различные сложные модели освещения и поверхностей.

*Дальше* я проводил сравнение с помощью функции `psnr`. (*пиковое отношение сигнала к шуму*). Данная метрика используется для сравнения зашумленного изображения и чистого. На выходе получается положительное число; чем оно больше, тем больше изображения похожи друг на друга.

с помощью реализованной в языке питон функции я получил графики, о которых подробно далее

### **Слайд “Реалистичный рендеринг”**

Реалистичный рендеринг позволяет создавать абсолютно любые формы и делать так, чтоб они выглядели как настоящие.

(Как говорилось ранее) эти программы применяются в кино- и телеиндустрии, дизайне, архитектуре и т.п. На данных изображениях можно видеть результат работы каждого из исследуемых рендеров. Как можно видеть на первом рисунке отрисованы различные поверхности, такие как стекло, пластик, древесина, металл и тд. На втором рисунке используется технология подповерхностного рассеивания для реалистичного отображения кожи. На третьем можно увидеть поверхность воды, учитывающую закон Френеля, и множество источников освещения.

Но как все это работает?

### **Слайд “Концепция PBR”**

Концепция физически корректного рендеринга базируется на 2 вещах: алгоритмы трассировки и уравнение рендеринга. Уравнение рендеринга это интеграл, который просчитывает энергетическую яркость, это буква  $L$ , поверхности с учетом огромного числа факторов.  $L_e$  в данном уравнении это собственная энергетическая яркость поверхности, то грубо говоря, собственное излучение света.  $L_i$  это энергетическая яркость падающего луча из направления  $i$ . это направление определяется вектором “омега малое  $i$ ”. Омега большое, это полусфера, которая учитывает любое направление  $i$ .  $f_r$  это Двухлучевая (или двунаправленная) функция распределения рассеивания. Эта функция отвечает за то, как определенная функция реагирует на падающий свет: куда направлен отраженный луч, куда преломленный, какую дискретную информацию он оставляет - в общем эта функция является определяющей материала. На рисунке 4 вы можете видеть примеры различных поверхностей, полученных с помощью этой функции.

### **Слайд “Алгоритмы трассировки”**

Рассмотрим теперь вторую составляющую рендеринга. Цифровое изображение представляется в виде набора пикселей. Для того чтобы вычислить дискретные значения пикселей, необходимо взять за образец исходную непрерывно определяемую функцию изображения. В PBR единственный способ получить информацию о функции изображения – это взять ее образец с помощью трассировки лучей. Изображение может быть создано только путем выборки значений функции в конкретных позициях пикселей. Чем больше выборок для разных пикселей, тем лучше качество.

**Первый алгоритм** - самый простой и самый ресурсоемкий. Он был описан в той же статье 86 года, где впервые было описано уравнение рендеринга. Этот алгоритм заключается в симулировании пути света начиная от наблюдателя. После столкновения с поверхностью в псевдослучайном направлении выпускается отраженный луч и преломленный (он же и теневой). В

каждой точке столкновения вычисляется, так называемая оценка монте-карло - ряд, который приближенно вычисляет значение уравнения рендеринга. Отражение происходит до полного поглощения луча, причем можно искусственно ограничить количество отражений, используя метод русской рулетки.

Второй алгоритм - Метод фотонных карт. Он состоит из двух этапов. На первом этапе световые лучи, которые содержат некоторую дискретную информацию, выпускаются из источников освещения и для каждой точки на поверхности сохраняется информация об ее освещенности в кэшэ.

На втором этапе лучи выпускаются уже из камеры и при попадании на поверхность информация об освещении собирается в некоторой окрестности точки падения.

Этот алгоритм содержит проблему кэша, когда кол-во информации может в нем не поместиться. В рендерах реализован модифицированный алгоритм, который называется стохастическое прогрессивное фотонное картирование, в котором во-первых первый и второй этапы меняются местами, а во вторых используется более экономная модель.

Третий алгоритм - двунаправленная трассировка пути, в котором строится сначала путь из камеры, потом из источника света и потом они соединяются. Такой способ позволяет сэкономить вычислительные ресурсы, потому что каждый луч вносит существенный вклад в изображение.

### **Слайд “Сцены для сравнения рендеров”**

Первый эксперимент - модель стакана с стеклянной поверхностью, в основном исследуется способность рендеров и алгоритмов вычислять каустики.

во втором эксперименте достаточно простая модель, в основном исследуется реалистичность. Так же здесь достаточно сложные пути света из-за преломления стеклянной поверхности бутылок.

в третьем эксперименте исследуется способность симулирования пропускающей свет ткани.

в четвертом исследуется способность рендеров реалистично симулировать поверхность жидкости и всевозможные световые явления, проявляющиеся при этом. В данной сцене существует два источника света: над и под жидкой поверхностью.

пятый эксперимент является наиболее сложным для алгоритмов рассиривки. это модель световода с двумя прямыми углами. индекс преломления стекла = 1.8.

### **Слайд “Результаты”**

В первом эксперименте  
разрешение 1920x1080

0. **В appleseed** используется алгоритм SPPM

Для обсчета прямого освещения используется SPPM

Кол-во испускаемых лучей из источника: 10M;

Радиус окрестности: 0.1;

Начальное кол-во отскоков возможных отражений до ограничения семплирования с помощью русской рулетки: 6

Кол-во итераций (семплов): 25, 52, 156, 370.

1. **Cycles**

Используется алгоритм Path Tracing.

Максимальное кол-во отражений/преломлений лучей испускаемых от камеры от:

Диффузной поверхности – 780;

Глянцевой поверхности – 9750;

Пропускающей поверхности – 7800;

Кол-во итераций (семплов): 17, 32, 89, 195.

2. **B LuxCore**

Используется алгоритм Path Tracing + Caustic light cache;

Кол-во испускаемых лучей из источника: 10M;

Радиус окрестности: 0.01;

Максимальное кол-во отражений фотона, выпущенного из источника света: 16;

Максимальное кол-во отражений лучей, испускаемых от камеры от:

Диффузной поверхности – 16;

Глянцевой – 36;

Зеркальной – 38;

В первых трех графиках отображено процентное соотношение увеличения скорости рендеринга при увеличении кол-ва потоков вдвое: с 6ти до 12ти. Можно заметить что все рендеры увеличивают свою скорость примерно на 70%, кроме cycles, у которого 60%.

На рисунках 4-6 отображено соотношения схожести с эталонным изображением. Как видно appleseed и luxcore практически одинаково отображают 1 модель. это связано с тем, что в них используются похожие алгоритмы. В cycles же используется трассировка пути, которой сложно даются вычисления каустик.

Во втором эксперименте  
разрешение изображения 2000x1700

#### 1. appleseed

используется трассировка пути.

Максимальное кол-во отражений/преломлений лучей испускаемых от камеры от:

Диффузной поверхности – 8;

Глянцевой поверхности – 8;

Пропускающей поверхности – 8;

Кол-во итераций 6 потоков (семплов): 7, 14, 27, 54.

#### 2. cycles

трассировка пути

Максимальное кол-во отражений/преломлений лучей испускаемых от камеры от:

Диффузной поверхности – 16;

Глянцевой поверхности – 6;

Пропускающей поверхности – 8;

Кол-во итераций 6 потоков (семплов): 36, 68, 72, 139.

#### 3. luxcore

трассировка пути

Диффузной поверхности – 4;

Глянцевой(отражающей) поверхности – 12;

Пропускающей поверхности – 12;

Кол-во итераций 6 потоков (семплов): 88, 173, 335, 676.

Из данного эксперимента касательно скорости можно сделать аналогичный с прошлым экспериментом вывод.

Из данных графиков видно, что appleseed показывает наибольшую схожесть, но на такой результат влияет то, что в качестве эталонного изображения взят

результат модели того же appleseed но за неограниченное время. Здесь важно обратить внимание на скорость роста графиков. Как можно видеть, наименьшая скорость роста у рендера luxcore. это говорит о том, что этим рендером достигнут пик качества, когда последующие итерации не вносят больших изменений в результирующее изображение. Аналогичный результат показывает и рендер cycles.